Some distance properties of convolutional codes.

Alfredson, Leonard Eric

Monterey, California. Naval Postgraduate School

SOME DISTANCE PROPERTIES OF CONVOLUTIONAL CODES

Leonard Eric Alfredson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

Some Distance Properties of Convolutional Codes

by

Leonard Eric Alfredson

Thesis Advisor: J. M. Geist

March 1972

Some Distance Properties of Convolutional Codes

by

Leonard Eric Alfredson
Lieutenant, Civil Engineer Corps, United States Navy
B.S., Northwestern University, 1964

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
March 1972

ABSTRACT

Various representations of convolutional codes useful in analyzing distance properties are presented. Row distance, column distance, minimum distance, and free distance are defined. Known bounds on these distances are summarized, and where instructive, the methods of proof are indicated.

A novel approach to the distance structure of a code is given in the form of a plot of row distance and column distance against depth into the code trellis. Bounds on minimum distance are applied to determine behavior of row and column distance.

Finally, the problem of determining the length of sequence necessary to produce the minimum weight codeword is considered. A bound for systematic codes is presented. This bound appears to be the tightest bound on this length presently known.

TABLE OF CONTENTS

LIST OF FIGURES

I. CONVOLUTIONAL CODING

A. INTRODUCTION

A general convolutional encoder is shown in Figure 1. At each instant of time, an information symbol is present on each of the k input lines, and an encoded symbol is present on each of the n output lines ($n \geqq k$) . The encoded symbols depend not only upon the information symbols present at that time but also upon the information symbols present during the previous m units of time. A code thus described is called an (n,K) convolutional code and is said to be of rate $R = k/n$ and to have memory m, and constraint length m + 1 blocks or (m + 1)n symbols.

In general, the symbols are elements of a finite field GF(q). This paper considers the binary case of symbols in GF(2), and all operations on symbols shall be assumed to be carried out in GF(2) unless otherwise stated.

B. POLYNOMIAL REPRESENTATION

Using the delay operator notation where D^j represents a delay of j bit-times the input sequence into the jth input line may be represented as a formal power series $X^j(D)$.

$$X^{(j)}(D) = x_0^{(j)} + x_1^{(j)}D + x_2^{(j)}D^2 + \ldots,j = 1,2,\ldots.k$$

Similarly, the output sequence from the jth output line is

$$Y^{(j)}(D) = y_0^{(j)} + y_1^{(j)}D + y_2^{(j)}D^2 + \ldots,j = 1,2,\ldots,n.$$

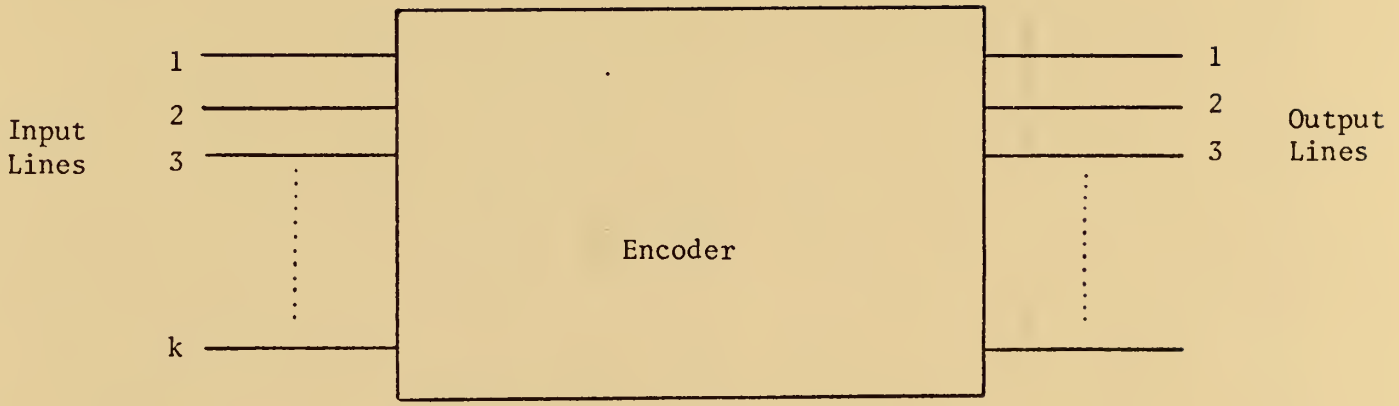


Figure 1. General Convolutional Encoder.

Each output sequence is a linear combination of digits in the input sequences, and therefore can be represented as

$$Y^{(j)}(D) = G^{(j)}(D)X^{(1)}(D) + H^{(j)}(D)X^{(2)}(D) + \ldots + Z^{(j)}(D)X^{(k)}(D).$$

In general the set $\left\{G^{(j)}(D) , \ldots, Z^{(j)}(D) ; j = 1,2,\ldots n\right\}$ consists of nk polynomials known as code-generating polynomials. The highest power of D appearing in these polynomials is m. Specifying the code-generating polynomials completely specifies a convoluntional code.

Suppose it is desired to specify a (2,1) convolutional code with m = 3. Then 2 x 1 polynomials with degree 3 at most are needed.

$$G^{(1)}(D) = g_0^{(1)} + g_1^{(1)}D + g_2^{(1)}D^2 + g_3^{(1)}D^3$$

$$G^{(2)}(D) = g_0^{(2)} + g_1^{(2)}D + g_2^{(2)}D^2 + g_3^{(2)}D^3$$

If we assign a value (either 0 or 1) to each $g_j^{(k)}$, a code is specified. That is, for any given input sequence one could then determine precisely the output sequence on each of the two output lines.

C. SHIFT REGISTER REPRESENTATION

A binary convolutional code can be represented (and implemented) using bit-time delay elements and modulo-2 adders. Figure 2 shows the configuration of an encoder for the previous example. Each symbol $\oplus$ represents a GF(2) addition. Wherever a $g_j^{(k)}$ appears, it represents a connection if $g_j^{(k)} = 1$ and no connection if $g_j^{(k)} = 0$. Since the code generating polynomials determine the pattern of connections, the shift register encoder also completely specifies a code. Specifying shift register connections and specifying the generator polynomials are completely equivalent.

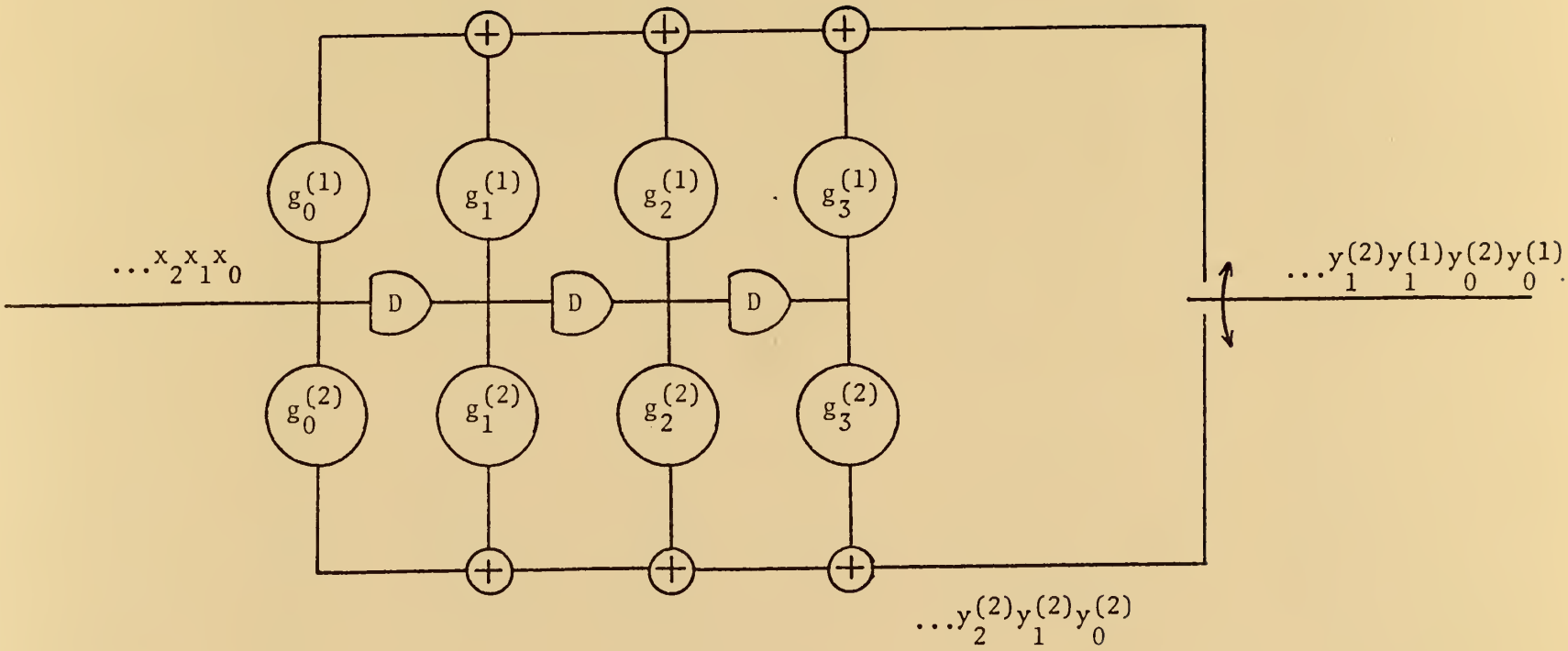


Figure 2. Representation of shift register encoder for R=1/2, m=3 code.

Figure 2 also shows a sampler at the output of the encoder. The purpose of the sampler is to combine the output symbols on all of the output lines into a single output sequence, Y. It does this by sequentially sampling all output lines once during each delay period. For the example, the output sequence of the encoder becomes

$$Y = y_0^{(1)} \; y_0^{(2)} \; y_1^{(1)} \; y_1^{(2)} \; y_2^{(1)} \; y_2^{(2)} \; \cdots$$

where the superscript indicates the line being sampled and the subscript is the time interval during which the sample is taken. When using the polynomial representation, the output sequence of the encoder can be obtained by interlacing the symbols of the n output line sequences.

D. MATRIX REPRESENTATION

A convolutional encoder may also be described by a generator matrix. Lin [1] describes in detail the construction of the generator matrix for a general binary convolutional code; this paper is confined to a description of how the matrix is used and an illustration of the generator matrix for the continuing example.

If an input sequence is represented as a semi-infinite vector and the generator matrix as $\underline{G}_\infty$, then $\underline{G}_\infty$ has the property

$$\underline{X}\,\underline{G}_\infty = \underline{Y}$$

where $\underline{Y}$ is the vector representation of the output sequence of the encoder. For the example,

$$\underline{G}_\infty = \begin{bmatrix} g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & g_3^{(1)} & g_3^{(2)} & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & g_3^{(1)} & g_3^{(2)} & 0 & 0 & 0 & \cdots & \\ 0 & 0 & 0 & 0 & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & g_3^{(1)} & g_3^{(2)} & 0 & 0 & \cdots \\ \vdots & & & & & & & & & & & & & & \end{bmatrix}$$

It is easily verified that the matrix operation gives the same result as using the polynomials and interlacing the output symbols. It is, in fact, only a different way of describing exactly the same operations. It is included, however, because it will give another perspective when looking at some of the distance properties of convolutional codes.

E. ADDITIONAL DEFINITIONS

A convolutional code may be either systematic or nonsystematic. A systematic code is one for which the first k of every n transmitted symbols are the k information symbols being encoded. Then each group of n symbols may be considered as a code "word" with k information bits and n - k check bits. It should be noted, however, that unlike block codes the check bits depend on the previous mk information bits as well as the information bits in the word.

In the example, the code is systematic if $G^{(1)}(D) = 1$. In shift register form, the code is systematic if the first k output lines each are connected directly to the respective input lines and nothing else. In Figure 2, the code is systematic if $g_0^{(1)} = 1$ and $g_1^{(1)} = g_2^{(1)} = g_3^{(1)} = 0$.

Nonsystematic codes may be catastrophic error propagating. A code is catastrophic if there exists an input sequence of infinite weight which is encoded into an output sequence which has finite weight. Massey and Sain [2] prove a theorem which states that a rate 1/n convolutional code is free of catastrophic error propagation if and only if the greatest common divisor of the n generator polynomials is D^L for some integer value of L. Rosenberg [3] develops some properties of noncatastrophic codes which will be helpful when discussing distance properties of noncatastrophic codes.

II. DISTANCE MEASURES

A. COLUMN DISTANCE

Since convolutional codes are linear codes, the minimum distance between output sequences is also the minimum Hamming weight of a non-zero output sequence. The distance between sequences is a function of the length of the sequences being considered. Column distance, d_j, is defined as the Hamming weight of the minimum weight output sequence generated by all input sequences of j + 1 symbols which begin with a non-zero symbol. For a rate 1/n code, d_j is found by the following procedure in polynomial representation:

$$Y^{(k)}(D) = G^{(k)}(D)\,\hat{X}(D) \quad , \; k = 1,2,\ldots,n$$

where

$$\hat{X}(D) = x_0 + x_1 D + x_2 D^2 + \ldots + x_j D^j \quad , \quad x_0 \neq 0 \; .$$

The next step is to truncate $Y^{(k)}(D)$ by setting all coefficients of powers of D greater than j equal to zero.

$$\hat{Y}^{(k)}(D) = y_0^{(k)} + y_1^{(k)} D + \ldots + y_j^{(k)} D^j$$

Then

$$d_j = \underset{\text{all } \hat{x}(D)}{\text{minimum}} \left\{ \sum_{k=1}^{n} W_H\left(\hat{Y}^{(k)}(D)\right) \right\}$$

where W_H is the Hamming weight operator.

One may also obtain d_j directly from the matrix representation. The method here is to form a truncated version of the semi-infinite generator matrix $\underline{G}_\infty$. The truncated matrix, called $\underline{G}_j$, is a matrix consisting of the

first $n(j + 1)$ elements of the first $(j + 1)$ rows of $\underline{G}_\infty$. Then d_j is the minimum weight of a linear combination of rows (which includes the first row) of $\underline{G}_j$. For the example in the previous chapter, the form of a generator matrix for a rate 1/2, m = 3 code was shown. Then to find d_2, one forms the matrix

$$\underline{G}_2 = \begin{bmatrix} g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} \\ 0 & 0 & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} \\ 0 & 0 & 0 & 0 & g_0^{(1)} & g_0^{(2)} \end{bmatrix}$$

which contains $j + 1$ rows and $n(j + 1)$ columns. Consider all possible linear combinations of rows which include the first row. The minimum Hamming weight which results is d_2.

If the above matrix had the values

$$\underline{G}_2 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

then $d_2 = 3$ which is the weight of the combination of all three rows.

The above procedure is a shorthand method of finding the minimum weight output sequence, since each different linear combination of rows corresponds to the multiplication of an input vector and the generator matrix. Since only the weights are important at this time, and not the actual sequences, it is not necessary to examine the multiplication process which would yield each combination of rows.

Before looking at d_j in terms of a shift register encoder, the encoder must be shown as a finite state device with 2^m possible states depending on the contents of the shift register. A trellis diagram [4] is a method of displaying the possible state transition trajectories. Figure 3 shows a simple encoder and its trellis diagram. Each node corresponds to a state of the shift register, and each branch to a transition from one state to another. The digits labeling each branch are the output symbols from the encoder produced by the transition. Column distance, d_j, is the minimum Hamming weight of a path $j + 1$ branches long which begins with the transition from state 0...00 to state 10...0.

B. ROW DISTANCE

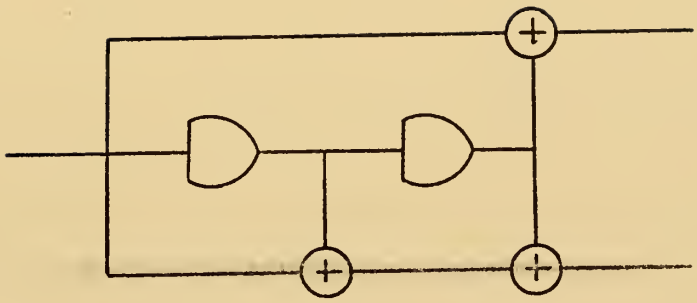
Row distance, r_j, is the Hamming weight of the minimum weight output sequence which can be generated by an input sequence of $j + 1 + m$ symbols which begin with a non-zero symbol and end with m zeros. In polynomial notation for a rate 1/n code, the expression becomes

$$r_j = \underset{\text{all } \hat{X}(D)}{\text{minimum}} \left\{ W_H \left[G^{(1)}(D)\, X(D) \right] + \ldots + W_H \left[G^{(n)}(D) X(d) \right] \right\}$$

where

$$\hat{X}(D) = x_0 + x_1 D + x_2 D^2 + \ldots + x_j D^j \quad , \quad x_0 \neq 0.$$

In matrix notation a truncated version of $\underline{G}_\infty$ which we will call $\hat{\underline{G}}_j$ is formed. $\hat{\underline{G}}_j$ is composed of the first $n(j + 1 + m)$ elements of the first $j + 1$ rows of $\underline{G}_\infty$. For the m = 3 rate 1/2 code, the truncated matrix for finding r_2 is



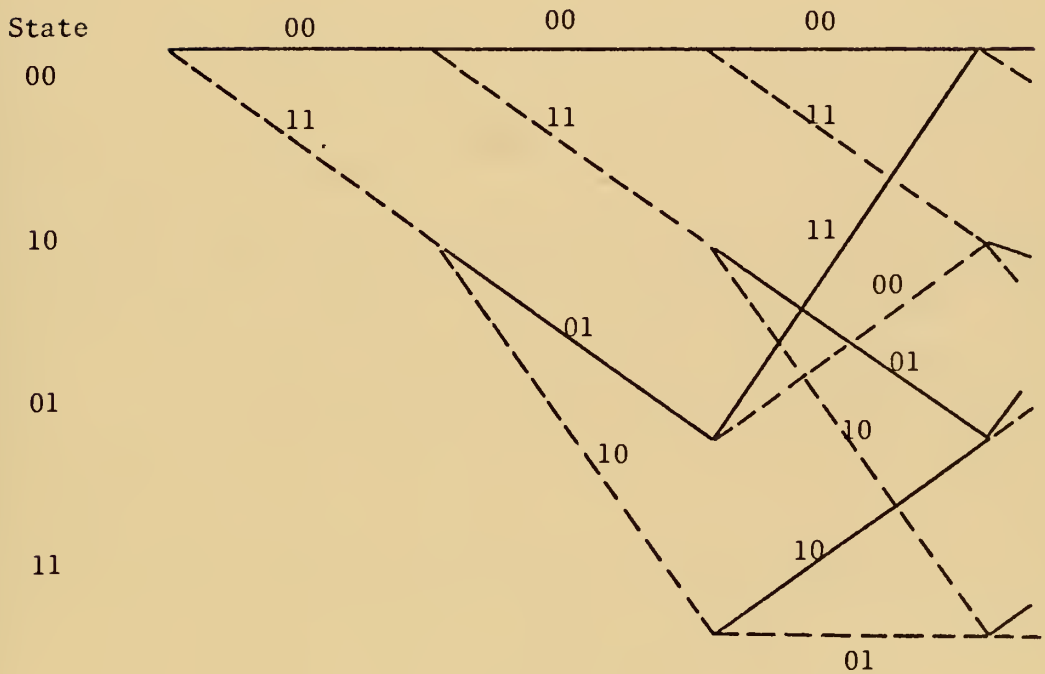


Figure 3. A rate 1/2, m=2 encoder and associated trellis diagram.

$$\hat{\underline{G}}_2 = \begin{bmatrix} g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & g_3^{(1)} & g_3^{(2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & g_3^{(1)} & g_3^{(2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & g_3^{(1)} & g_3^{(2)} \end{bmatrix}$$

As before, one examines all linear combinations of rows which include the first. The minimum of the weights thus obtained is r_j.

If the values of the elements of the matrix above were

$$\hat{\underline{G}}_2 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

then $r_2 = 6$ which is the weight of the first row and also the weight of the combination of all three rows.

To use the trellis diagram to determine r_j for a code, the minimum weight of "eligible" paths of lengths $j + 1 + m$ branches must be found. To be eligible, the first branch of the path must be a transition from state 00...00 to state 10...0, and the final m branches must be transitions caused by zero input symbols.

Column distance and row distance are very general measures of minimum distance between output sequences. There is no general way to express d_j or r_j in algebraic form. When attempting to evaluate d_j and r_j for a given code the difficulty grows exponentially with j. It is easily seen that the methods for finding d_j and r_j described above all consist of determining the weight of 2^j possible output sequences and then choosing the minimum.

As j becomes large, this method quickly becomes impractical even with the use of a digital computer. More specific distance measures are used when attempting to judge the relative "goodness" of convolutional codes for use with particular decoding algorithms. The two most common and important of these are defined below. This paper will later describe how these distances are bounded.

C. MINIMUM DISTANCE AND FREE DISTANCE

Minimum distance, d_{min}, refers to the minimum distance between initial code words. An initial code word is an output sequence of length $n(m + 1)$ symbols generated when the shift register is initially in the zero state. This says that $d_{min} = d_j$ for $j = m$. Minimum distance is an important measure for a code, when a decoding algorithm is used which bases decisions on encoded sequences one contraint length long. One such algorithm is Threshold Decoding [5] .

For decoding algorithms which make decisions based on much longer sequences such as Viterbi [6] decoding and sequential decoding, a reasonable measure of error correcting ability is free distance. Free distance is defined as

$$d_{free} = \lim_{j \to \infty} d_j \ .$$

It will be shown later that this limit exists. Since for reasons mentioned, this is difficult to obtain for a code in general, many theorists have attempted to find an upper limit L such that d_j for $j = L$ gives free distance for a code. This bound on computation required to find d_{free} is described later.

III. BOUNDS ON MINIMUM DISTANCE

A. GILBERT LOWER BOUND

Massey [5] proved a theorem which provides a lower bound on minimum distance for systematic convolutional codes which is equivalent to the Gilbert bound on minimum distance for block codes [7]. The following is a statement of Massey's theorem for a binary code, and a description of the method of his proof.

THEOREM: Given a rate $R = k/n$ and a constraint length $n_A = n(m + 1)$, then there exists at least one convolutional code with a minimum distance d, where d is the largest integer for which

$$\sum_{j=1}^{d-1} \binom{n_A}{j} < 2^{n_A(1-R)}$$

METHOD OF PROOF: A convolutional code has minimum distance d if it has no initial code word of weight d - 1 or less for which some first information symbol is non-zero. Massey computes the total number of non-zero initial code words which have wieght d - 1 or less and multiplies by the number of distinct codes in which each initial code word can appear. If this product is less than the total number of distinct convolutional codes, then a code must exist which has minimum distance d. The algebraic expression of this inequality is

$$\left[\sum_{j=1}^{d-1} \binom{n_A}{j}\right] \left[2^{(m+1)(n-k)(k-1)} \right] < 2^{(n-k)(k)(m+1)}$$

which reduces to

$$\sum_{j=1}^{d-1} \binom{n_A}{j} < 2^{N_A(1-R)}$$

Then, using the well known inequality

$$\sum_{j=0}^{\delta n} \binom{n}{j} \leqq 2^{nH(\delta)}, \quad \delta \leqq 1/2$$

the bound may be expressed as

$$H\left(\frac{d}{n_A}\right) \leqq 1 - R$$

where

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x).$$

Although Massey proves the bound for systematic codes, it is not difficult to extend the proof to include non-systematic codes, where "non-systematic" is used in the sense which excludes systematic codes, rather than in the sense which includes systematic codes as a special case. For systematic codes, choosing an initial code word specifies the information sequence, and the proof proceeds to deal with the $(n - k)k$ parity bit polynomials. For a non-systematic code one need only specify the first k^2 of the required nk polynomials. Then an initial code word is assumed. This word and the k^2 polynomials uniquely determine the input polynomials. The proof then proceeds in the same fashion as for systematic codes except each side of the inequality must be multiplied by $2^{k^2(m + 1)}$ to allow for the number of ways the first k^2 polynomials may be specified. Since this factor appears on both sides of the inequality, the final

expression for the Gilbert bound remains the same. Thus if the Gilbert bound inequality is satisfied, it guarantees the existence of both a systematic and a non-systematic convolutional code with minimum distance of at least d. This bound only guarantees the existence of such codes, but does not reveal a procedure for finding them.

B. PLOTKIN UPPER BOUND

Massey [8] has also derived an upper bound on minimum distance for convolutional codes in a manner similar to the Plotkin bound for block codes. This section presents the form of the bound and omits the method of proof. Described later is the method of proof for another upper bound which yields a tighter bound for systematic codes, and the same bound for non-systematic codes as Massey's version. The Plotkin bound states that the minimum distance of a binary convolutional code satisfies the inequality

$$d_{min} \leq \left[\frac{m+5}{2}\right] (n-k) + 1$$

where R = k/n. For a rate 1/n code, this inequality reduces to

$$\frac{d_{min}}{n_A} \leq \frac{1}{2} (1-R) .$$

C. MCELIECE-RUMSEY UPPER BOUND

McEliece and Rumsey [9] derived an upper bound on minimum distance for systematic convolutional codes of rate 1/n. The statement of this bound is as follows:

THEOREM: For each integer $h \geq 1$,

$$d_{min} \leqq \frac{2^h}{2^h - 1} \left(\frac{hn}{2} + \frac{m(n-1)}{2} \right)$$

METHOD OF PROOF: McEliece and Rumsey demonstrate that a non-zero output polynomial generated by a non-zero input of length h is at least as likely to have a zero for the i^{th} coefficient as it is to have a one for its i^{th} coefficient. The right-hand side of the inequality thus is an expression for the average weight of a non-zero code word caused by an input of length h, and the inequality results from the fact that the minimum weight can be no greater than the average weight. A corollary is presented which states if

$$\frac{(n-1)(m+1)}{1 + \log_2 \left[(n-1(m+1) \right]} > n$$

then

$$d_{min} < \frac{(n-1)(m+1)}{2} + \frac{n}{2} \log_2 \left[(n-1)(m+1) \right] + 1$$

Asymptotically this can be stated as

$$\lim_{m \to \infty} \frac{d_{min}}{n_A} < \frac{1}{2} (1-R)$$

Costello [10] has generalized the McEliece-Rumsey bound to include all fixed (non-time varying) convolutional codes. The asymptotic form of this general bound is

$$\lim_{m \to \infty} \frac{d_{min}}{n_A} < \frac{1}{2} .$$

D. SUMMARY

Figure 4 shows plots of the asymptotic forms of the bounds described in this chapter [10]. The bounds are asymptotic because m is assumed to be arbitrarily large. It must be remembered that for a given rate, at least one code exists which has a ratio d/n_A equal to or greater than the Gilbert bound. No code exists which has a greater d/n_A ratio than indicated by the appropriate upper bound. Nothing is known about how to construct codes which are guaranteed to be in this region. For smaller values of m, on the order of m = 20, codes have been formed [11] which have minimum distances at or near the appropriate upper bound.

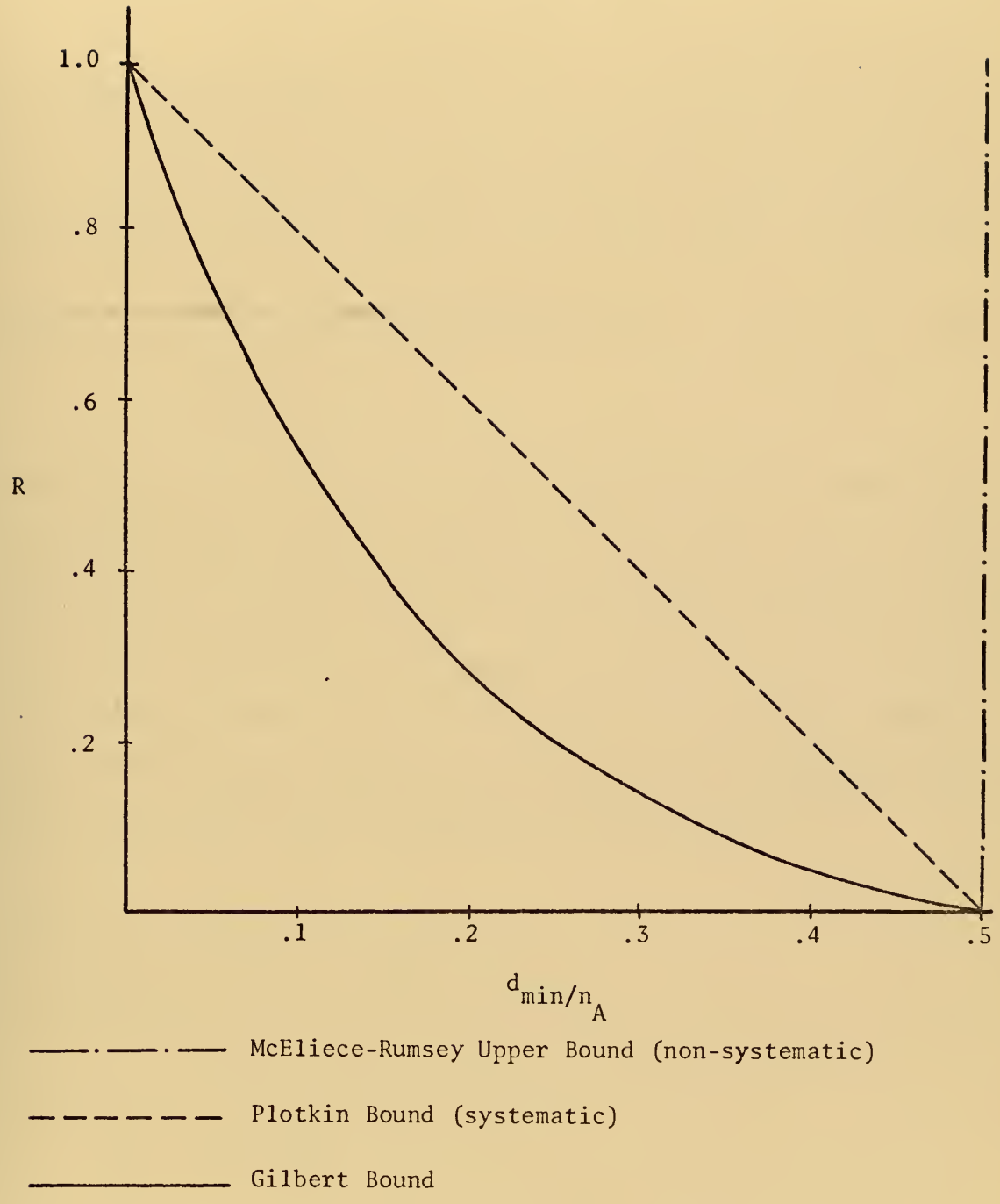


Figure 4. Asymptotic bounds on minimum distance.

IV. BOUNDS ON FREE DISTANCE

A. PROPERTIES OF d_j AND r_j

It is necessary to establish some properties of column distance and row distance which will be used when discussing bounds on free distance.

Property 1: r_o = weight of the generator $\leqq n_A$

$$0 \leqq d_o \leqq n$$

An input sequence consisting of a one followed by all zeros generates the output sequence which has weight r_o. It is obvious that this weight is exactly equal to the total weight of the generator polynomials and that this total weight cannot exceed $(m + 1)n$. The weight of the first branch from the all-zero path is d_o, and since any single branch has weight no greater than n, $d_o \leqq n$. Since d_o is maximum when all generators start with one, practical codes are universally chosen so $d_o = n$.

Property 2: $d_j \leqq d_{j+1}$, $j = 0,1,2,\ldots$

The general form of the truncated matrix $\underline{G}_{j+1}$ is

$$\underline{G}_{j+1} = \left[\begin{array}{c|ccc} & x & \cdots & x \\ \underline{G}_j & \vdots & & \vdots \\ & & & \\ \hline 00\cdots00 & x & \cdots & x \end{array}\right]$$

where each of the elements labeled x may be zero or non-zero depending upon the particular code. Since all elements below the submatrix $\underline{G}_j$ are zero, it is easily seen that any linear combination of rows of $\underline{G}_{j+1}$ can have weight no less than the corresponding rows of $\underline{G}_j$ alone. Thus d_j is

non-decreasing in j. It should be recalled that these linear combinations are being used to determine weights of sequences which would actually be produced by multiplication of an input vector with the generator matrix.

Property 3: $r_j \geqq r_{j+1}$, $j = 0, 1, 2, \ldots$

The general form of $\underline{\hat{G}}_{j+1}$ is

$$\underline{\hat{G}}_{j+1} = \left[\begin{array}{c|c} \underline{\hat{G}}_j & \begin{matrix} 0 \cdots 0 \\ \vdots \quad \vdots \\ 0 \cdots 0 \end{matrix} \\ \hline 00\cdots xx \quad \cdots\cdots \quad xx \end{array}\right]$$

where, as above, each of the elements labeled x may be zero or non-zero. It can be seen that if any linear combination of rows of $\underline{\hat{G}}_{j+1}$ which include the row outside of $\underline{\hat{G}}_j$ have weight greater than a combination of rows inside $\underline{\hat{G}}_j$, then that combination is not minimum weight and is not used in determining r_{j+1}. Therefore, r_j is non-increasing in j.

Property 4: $r_j \geqq d_j$, $j = 0, 1, 2, \ldots$

Another form for $\underline{\hat{G}}_j$ is

$$\underline{\hat{G}}_j = \left[\begin{array}{c|c} \underline{G}_j & \begin{matrix} x \cdots x \\ \vdots \quad \vdots \\ x \ldots x \end{matrix} \end{array}\right] .$$

It is seen that no linear combination of rows of $\underline{\hat{G}}_j$ can have weight less than the combination of the same rows of $\underline{G}_j$. Therefore, $r_j \geqq d_j$ for all j.

Property 5: The limits $r_\infty = \underset{j\to\infty}{\text{limit}}\ r_j$ and

$d_\infty = \underset{j\to\infty}{\text{limit}}\ d_j$ exist, and $r_\infty \geqq d_\infty$.

Properties 1 through 3 indicate that $\{d_j\}$ is a non-decreasing integer valued sequence bounded from above and $\{r_j\}$ is a non-increasing integer valued sequence bounded from below. Thus the sequences approach limits. From Property 4, $r_\infty \geq d_\infty$.

Since d_j and r_j are integer valued, the sequences achieve their limiting values for some finite L_c and L_r such that $d_\infty = d_{L_c}$ and $r_\infty = r_{L_r}$. Costello has shown that for non-catastrophic codes, $r_\infty = d_\infty$. Note that this does not imply $L_c = L_r$.

It might be well to consider the properties of r_j and d_j in terms of a trellis diagram such as that in Figure 3. Column distance, d_j, is the weight of the minimum weight path through the trellis which is $j + 1$ branches long and has a 00...00 to 10...00 transition as its first branch. It can be seen that each path of $j + 1$ branches has a path of j branches as a prefix, so the minimum weight of the $j + 1$ branches must be at least as great as the minimum weight of a j branch path. Thus d_j is a non-decreasing on j.

Row distance, r_j, is the weight of the minimum weight path through the trellis which is $j + 1 + m$ branches long, begins with a 00...00 to 10...00 transition, and ends at a 00...000 state. One possible path for r_j is the path which yielded r_{j-1} plus a 00...00 to 00...00 transition. Thus r_j can never be greater than r_{j-1}.

Since all initially non-zero paths $j + 1 + m$ branches long ending at the zero state contain as prefixes all initially non-zero paths $j + 1$ branches long, r_j can never be less than d_j.

B. LOWER BOUNDS ON FREE DISTANCE

The properties shown in the previous section can be stated as properties of various bounds. An upper bound on r_j for any j is an upper bound on d_j for all j. A lower bound on d_j for j = L is a lower bound on d_j for j > L. It follows from this last statement that Gilbert lower bound for minimum distance is also a lower bound on free distances.

Neumann [12] has given a bound for non-systematic codes which states that there is at least one binary non-systematic code such that for large values of m

$$\frac{d_{free}}{n_A} \geqq \begin{cases} 2H^{-1}(1-R) & \text{for } R \quad 0.37 \\ \dfrac{2R(1-2^{2R-1})}{H(1-2^{2R-1})+2R-1} & \text{for } R \quad 0.37 \end{cases}$$

A sketch of this bound is included in Figure 5. An interesting thing to note is that since the Neumann bound is above the upper bound for systematic codes for some range of R, then in that range of R there exists some non-systematic code with larger free distance than any systematic code.

C. UPPER BOUND ON FREE DISTANCE

An upper bound on minimum distance is not, in general, an upper bound on free distance. However, the McEliece-Rumsey bound, due to its method of proof, is actually a bound on r_j and therefore a bound of d_j for all j. This could be expressed as

$$r_j \leqq \frac{2^j}{2^j-1}\left[\frac{jn}{2} + \frac{m(n-1)}{2}\right] .$$

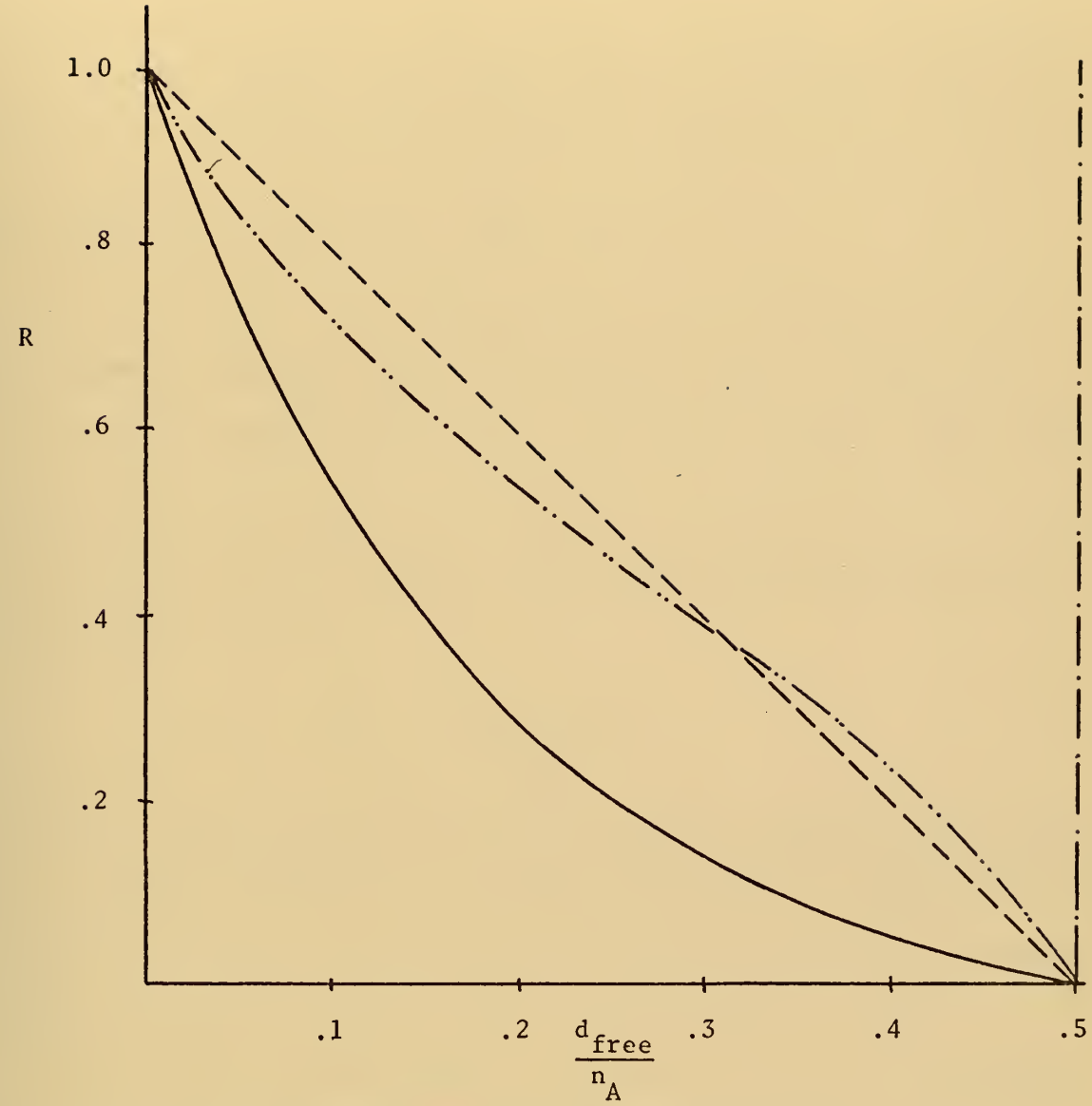


Figure 5. Asymptotic bounds on free distance.

The corollary used as a bound on column distance corresponds to the minimum of the right-hand side of the above inequality. The bound on r_j is only of interest in the region where it is decreasing, and as a bound on d_j, of concern only at a minimum point.

D. PERMISSIBLE REGIONS FOR d_j AND r_j WHEN $R = \frac{1}{2}$

Massey's Plotkin bound on d_{min} may be used to bound d_j for $j \leqq m$. This bound is the upper bound on the minimum weight of a code word $m + 1$ branches long produced by an encoder with m delay elements. Since every code sequence shorter than $m + 1$ branches is a prefix of an $m + 1$ branch length sequence, Massey's proof is valid while considering the first j delay elements. Thus

$$d_j \leqq 3 + \frac{j+1}{2} \quad , \; j \leqq m, \; R = \frac{1}{2} \quad .$$

Savage [13] has shown that there exist codes for which d_j is lower bounded by a line from the origin to the Gilbert bound at $j = m$. When these bounds are plotted together with the McEliece-Rumsey bound, the resulting permissible region for $\{d_j\}$ is as indicated by the shaded region of Figure 6.

Figure 7 shows typical forms of $\{r_j\}$ and $\{d_j\}$ for a non-catastrophic code. The rates of change of r_j and d_j are not known in general, do not approach a limit at the same rate, and may not change uniformly with j.

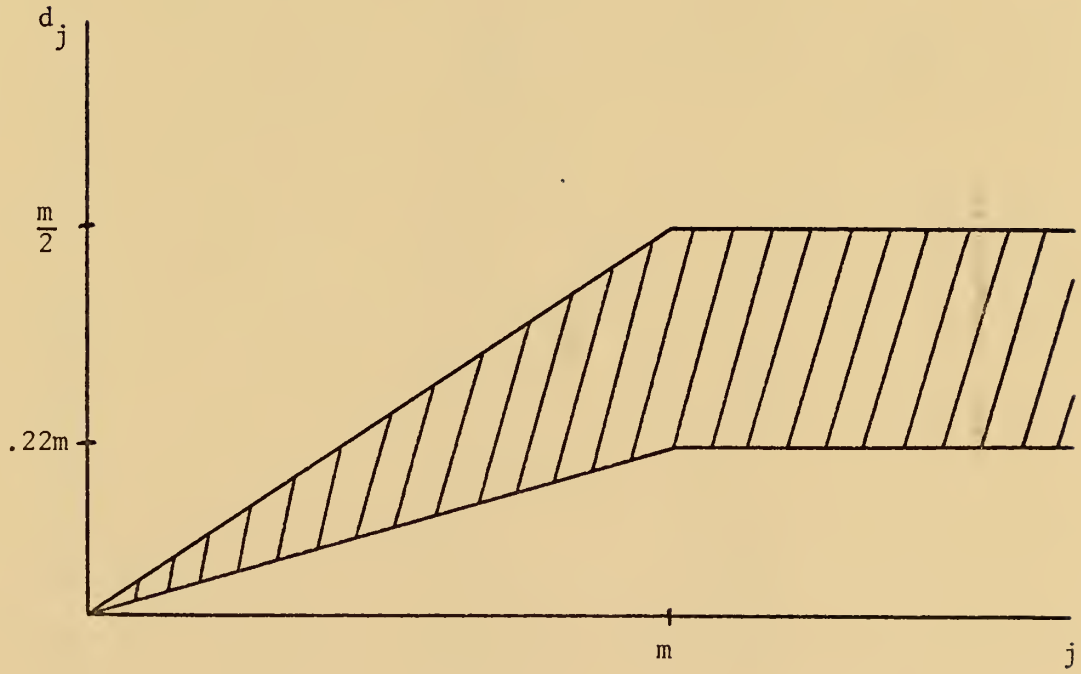


Figure 6. Permissible region for d_j. (Rate = $\frac{1}{2}$)

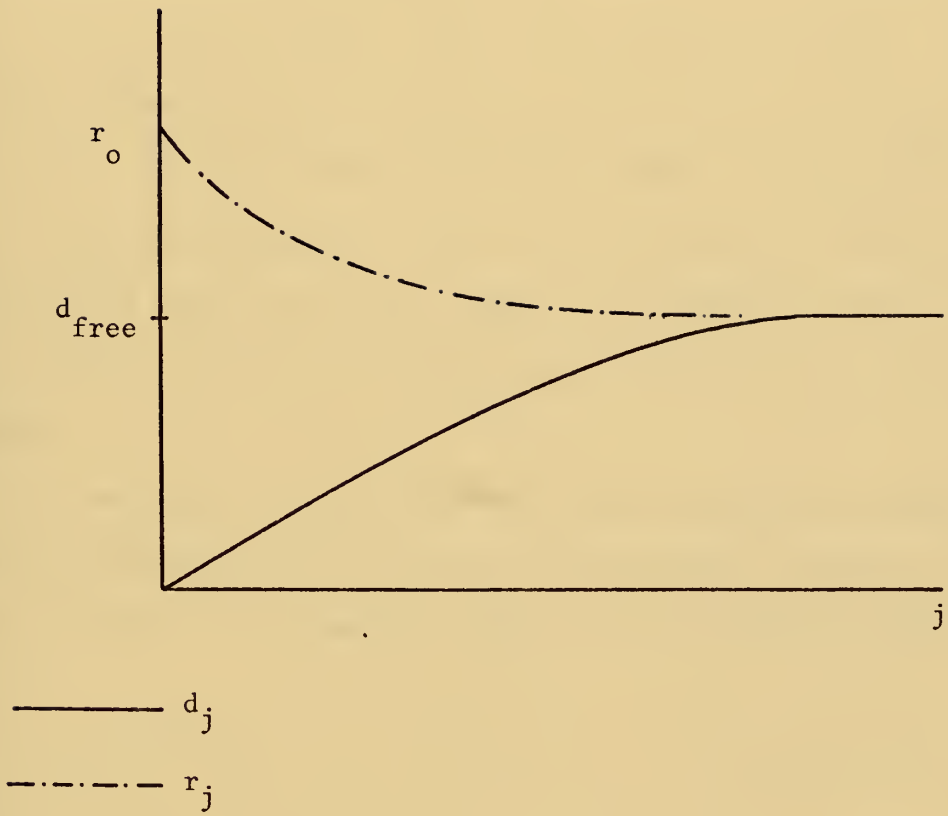


Figure 7. Typical behavior of r_j and d_j.

V. BOUNDS ON LENGTH OF OUTPUT SEQUENCE PRODUCING FREE DISTANCE

A. INTRODUCTION

Even after bounding free distance, the problem of determining the length of the output sequence which will yield d_{free} remains. This is an important parameter because between two codes with equal free distance, the code for which the smaller value of j yields $d_j = d_{free}$ has the better error correcting capability. Also, if it is desired to compute the free distance of a particular code, a bound on the length of paths to be searched is a valuable measure of the maximum amount of computation which will be required.

It is necessary to state explicitly some properties of non-catastrophic rate 1/n codes which will be used in the discussion. First, it is apparent that r_o is equal to the Hamming weight of the generators. That is,

$$r_o = W_H\left[G^{(1)}(D)\right] + W_H\left[G^{(2)}(D)\right] + \ldots + W_H\left[G^{(n)}(D)\right] .$$

Since the weight of the generators is at most n(m + 1),

$$r_o \leqq n(m + 1).$$

It should be recalled from the properties previously stated that r_o is an upper bound on d_j and r_j for all j. The second important property is the fact that a path through a trellis which yields r_j cannot leave the all-zero path once a zero state has been reached. Examination of the trellis shows any non-zero path which begins and ends at the all-zero state cannot touch

and depart from an all-zero node and still be a minimum weight path because any path departing from the all-zero path has weight at least as great as the all-zero path. Since m consecutive zeros in the input sequence drive the encoder to the all-zero state, the result is that any input sequence which determines the path yielding r_j cannot have an internal sequence of m consecutive zeros.

B. COSTELLO'S BOUND

Costello uses the fact that an input sequence with no more than m - 1 consecutive zeros cannot produce an output with more than (2m - 2) consecutive zero-weight output branches for a rate $\frac{1}{2}$ code. Then at least one non-zero element must be produced in every (2m - 1) encoded branches. Therefore at least 2(m + 1) ones must be produced in an output sequence of (2m - 1)2(m + 1) + 1 branches. Since this is the upper limit of r_o and therefore d_{free}, then

$$d_{free} = r_{4m^2 + 2m-1} .$$

Costello states that a similar argument can be used to show that for a systematic rate k/n code,

$$d_{free} = r_{(n - k)(m + 1)m}.$$

C. AN IMPROVED BOUND

For rate $\frac{1}{2}$ codes a better bound than Costell's can be derived by using tighter constraints. For the upper limit of d_{free}, use the McEliece-Rumsey bound rather than the maximum weight of the generator. This bound for a non-systematic rate $\frac{1}{2}$ code is

$$d_{free} < m + \log_2(m) + 1 .$$

A theorem by Odenwalder states that for a non-catastrophic rate 1/n code, the maximum number of consecutive zero weight branches which may occur is m - 1. This yields

$$d_{free} \leq r_{\lfloor m + \log_2(m-1) + 1 \rfloor} (m - 1) + 1 .$$

While this is an improvement over Costello's bound, it is still on the order of m^2. This bound was also derived by Bahl and Jelinek [14].

D. AN IMPROVED BOUND FOR RATE 1/n SYSTEMATIC CODES

Figure 8 is a state diagram introduced by Viterbi [15]. The $\underline{0}$ node is split so entrance into the diagram is from the zero node and exit from the diagram is the zero node. A path through the diagram represents a departure from and return to the all-zero path in the trellis. A broken line represents a branch caused by a non-zero input and a solid line is a branch caused by a zero input. Since the code is assumed systematic, branches caused by a non-zero input will have a weight of at least one.

Any path which yields r_j is a path through the state diagram. Since r_j is a minimum weight, the path must be non-looping. If an integer L is found such that the minimum weight of a non-looping path L branches long through the state diagram is at least as great as an upper bound on free distance, then

$$r_L = r_\infty$$

and

$$r_L = d_{free} .$$

The path whose weight grows most slowly with length is generated by an input sequence consisting of a one followed by the maximum permitted number of zeros followed by a one followed by the maximum permitted number of zeros, etc. The length of the zero sequences will be limited by the

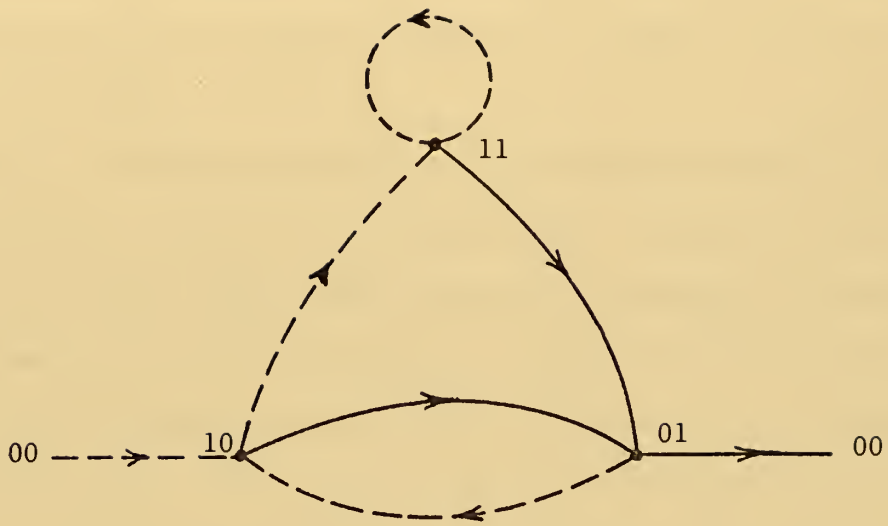


Figure 8. State diagram for m = 2 code.

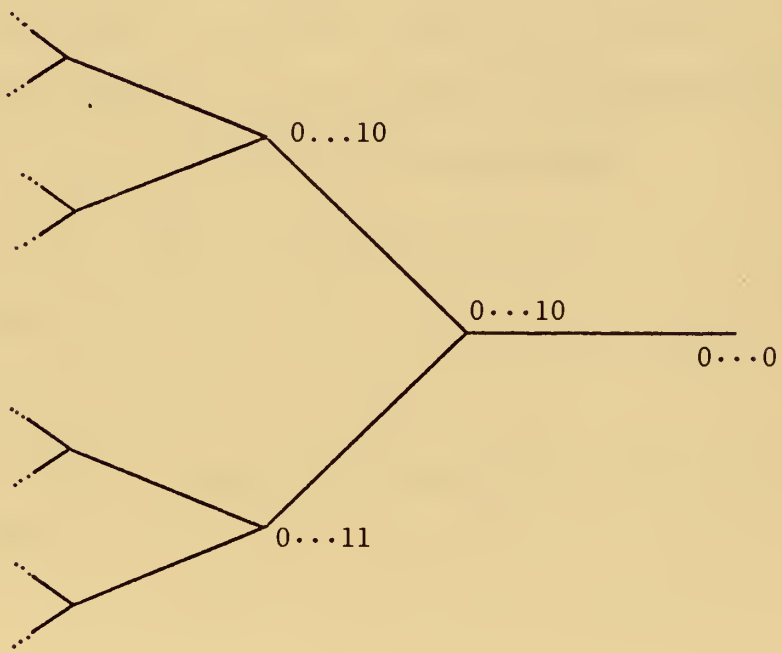


Figure 9. Tree of zero induced state transitions.

requirement that the path be non-looping, or the path shall not touch any node more than once. Minimum weight is assured by assuming a weight of one on branches caused by a non-zero input and a weight of zero on a branch caused by a zero input.

Figure 9 is a tree representation of all transitions caused by zero inputs. In general, the tree is m branches deep. A path of m zeros is reserved for the last m input symbols necessary to generate a total path $j + 1 + m$ branches long used to define r_j. Since any path of m zeros must touch the node to the left of the $\underline{0}$ node, a sequence of $m - 1$ consecutive zeros is not permitted. If it were, a node would be touched twice. Thus the first non-zero input may be followed by a sequence of at most $m - 2$ zeros. If nodes touched by such a sequence are removed from the tree, it is seen that two sequences of $m - 3$ zeros are permitted. Removing nodes touched by these sequences leaves four sequences of $m - 4$ zeros. Thus the first $m - 1$ output branches must have weight at least one, each of the next two $m - 2$ output branches must have weight at least one, and so on. Thus an output sequence is determined which has the following branch length:

$$L = m - 1 + 2(m - 2) + 4(m - 3) + 8\ (m - 4) + \ldots$$

The weight of this sequence must be at least

$$W_H = 1 + 2 + 4 + 8 + \ldots.$$

This can be put in the form of a summation and set greater than or equal to the McEliece-Rumsey bound.

$$\sum_{i=o}^{P} \left[2^i \right] + Q \geqq \frac{(n-1)(m+1)}{2} + \frac{n}{2} \log_2 \left[(n-1)(m+1) \right] + 1$$

where $Q < 2^{P+1}$. If the smallest integers which satisfy these relationships are called P* and Q*, then

$$L^* = \sum_{i=o}^{P^*} \left[2^i (m-i-1) \right] + Q^* (m-p^*-2) .$$

It follows that

$$d_{free} = r_{L^*} + 1$$

For n = 2, and m = 31, L* = 598 and $d_{free} = r_{599}$.

The Bahl-Jelinek bound reduced for systematic codes yields

$$d_{free} = r_{660} .$$

Costello's bound for a systematic code yields

$$d_{free} = r_{992} .$$

It is seen that the bound presented above is tighter than the Bahl-Jelinek bound, but the order of L* is still m^2 .

Miczo and Rudolph [16] examined the problem of bounding L* such that $d_{free} = d_{L^*}$ and demonstrated by counter-example that L* cannot be any constant multiple of m. They speculate that a bound of the form (m log m) might be placed on L*, but this has not been done.

VI. SUMMARY AND RECOMMENDATIONS

The various distance measures of convolutional codes have been defined, and the upper and lower bounds have been demonstrated. The bounds are based on the general structure of convolutional codes without considering the effect of specific code generating polynomials. In general, specifying generators does not provide a means of calculating distances since so little is known about the Hamming weight of products of polynomials. A logical extension of the topics in this paper is the study of codes whose generators have special properties. An examination of rate 1/2 complimentary codes by Bahl and Jelinek [11] is an example of such a study.

The bounds on the length of sequences which yield d_{free} appear to be subject to more tightening. The bound at the end of the previous chapter considered each branch to be of the minimum possible weight. An attempt to assign a more realistic distribution of weights to the branches of the state diagram might result in a tighter bound.

As mentioned earlier, the growth of d_j with j is an important property of codes, and narrowing the permissible region of d_j would be significant.

LIST OF REFERENCES

1. Lin, S., An Introduction to Error-Correcting Codes, p. 212-222, Prentice-Hall, 1970.

2. Massey, J.L. and Sain, M.K., "Codes, Automatic, and Continuous Systems: Explicit Interconnections," IEEE Transactions on Automatic Control, v. AC-12, p. 644-650, December 1967.

3. Rosenberg, W.J., Structural Properties of Convolutional Codes, Ph.D. Thesis, University of California, Los Angeles, 1971.

4. Forney, G.D., "Final Report on a study of a Simple Sequential Decoder," U.S. Army Satellite Communications Agency Contract DAAB07-68-C-0093, Appendix A, Codex Corporation, April 1968.

5. Massey, J.L., Threshold Decoding, p. 11-19, M.I.T. Press, 1963.

6. Viterbi, A.J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, v. IT-13, p. 260-269, 1967.

7. Peterson, W.W., Error-Connecting Codes, p. 48-52, M.I.T. Press, 1961.

8. Massey, J.L., "Some Algebraic and Distance Properties of Convolutional Codes," Error-Correcting Codes (e. H.B. Mann), Wiley and Sons, 1968.

9. McEliece, R. and Rumsey, H.C., "Capabilities of Convolutional Codes," Jet Propulsion Laboratory SPS 37-50, v. III, p. 248-251, 1968.

10. Costello, D.J., Construction of Convolutional Codes for Sequential Decoding, Ph.D. Thesis, University of Notre Dame, 1969.

11. Jelinek, F. and Bahl, L.R., "Rate $\frac{1}{2}$ Convolutional Codes With Complementary Generators," IEEE Transactions on Information Theory, v. IT-17, p. 718-728, November 1971.

12. Neumann, B., Distance Properties of Convolutional Codes, M.S. Thesis, Massachusetts Institute of Technology, 1968.

13. Savage, J.E., "Minimum Distance Estimates of the Performance of Sequential Decoders," IEEE Transactions On Information Theory, v. IT-15, p. 128-140, January 1969.

14. Bahl, L.R. and Jelinek, F., "On the Structure of Rate 1/n Convolutional Codes," IEEE Transactions on Information Theory, v. IT-18, p. 192-196, January 1972.

15. Viterbi, A.J., "Convolutional Codes and Their Performance in Communications Systems," IEEE Transactions on Communication Technology, v. COM-19, p. 751-772, October 1971.

16. Miczo, A. and Rudolph, L.D., "A Note on the Free Distance of a Convolutional Code," IEEE Transactions on Information Theory, v. IT-16, p. 646-648, September 1970.

INITIAL DISTRIBUTION LIST

1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Asst. Professor J.M. Geist, Code 52Gj Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor G.H. Marmont, Code 52Ma Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	LT Leonard Eric Alfredson, USN 3501 N. Main St. Rockford, Illinois 61103	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Some Distance Properties of Convolutional Codes

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; March 1972

5. AUTHOR(S) (First name, middle initial, last name)

6. REPORT DATE

March 1972

7a. TOTAL NO. OF PAGES

43

7b. NO. OF REFS

16

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

Various representations of convolutional codes useful in analyzing distance properties are presented. Row distance, column distance, minimum distance, and free distance are defined. Known bounds on these distances are summarized, and where instructive, the methods of proof are indicated.

A novel approach to the distance structure of a code is given in the form of a plot of row distance and column distance against depth into the code trellis. Bounds on minimum distance are applied to determine behavior of row and column distance.

Finally, the problem of determining the length of sequence necessary to produce the minimum weight codeword is considered. A bound for systematic codes is presented. This bound appears to be the tightest bound on this length presently known.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

A-31408

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Error-Correcting Codes						
Convolutional Codes						

134017
Thesis
A3754 Alfredson
c.1 Some distance properties of convolutional codes.